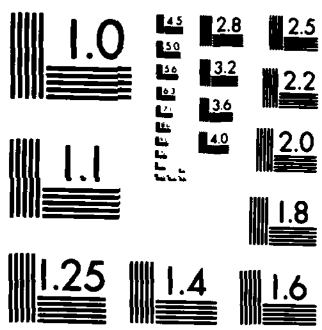
AD-A153 919 HIGH PERFORMANCE ASYNCHRONOUS LIMITED SENSING ALGORITHMS FOR CSMA AND CSM..(U) CONNECTICUT UNIV STORRS DEPT OF ELECTRICAL ENGINEERING AND CO.. 1/1

UNCLASSIFIED M GEORGIOPOULOS ET AL. MAR 85 F/G 12/1 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The University of Connecticut
SCHOOL OF ENGINEERING
Storrs, Connecticut 06268

"High Performance Asynchronous Limited Sensing Algorithms for CSMA and CSMA-CD Channels"

M. Georgiopoulos
L. Merakos
P. Papantoni-Kazakos

Technical Report UCT/DEECS/TR-85-2

March 1985

Department of
Electrical Engineering and Computer Science

"High Performance Asynchronous Limited Sensing Algorithms for CSMA and CSMA-CD Channels"

M. Georgiopoulos
L. Merakos
P. Papantoni-Kazakos

Technical Report UCT/DEECS/TR-85-2

March 1985

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): UCT/DEECS/TR-85-2	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 85-0398
6a. NAME OF PERFORMING ORGANIZATION: University of Connecticut; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code): Department of Electrical Engineering and Computer Science, Storrs CT 06268	7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR; 8b. OFFICE SYMBOL (If applicable): NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-83-0229
8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332-6448	10. SOURCE OF FUNDING NOS.: PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2304; TASK NO. A5; WORK UNIT NO.

11. TITLE (Include Security Classification): HIGH PERFORMANCE ASYNCHRONOUS LIMITED SENSING ALGORITHMS FOR CSMA AND CSMA-CD CHANNELS

12. PERSONAL AUTHOR(S): M. Georgiopoulos, L. Merakos and P. Papantoni-Kazakos

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	MAR 85	42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The authors consider the random multiple access of a collision-type, packet-switched channel, for the Poisson user model in a local area network environment, where "carrier sensing" techniques are possible due to small propagation delays.

The authors propose and analyze asynchronous (unslotted) random access algorithms that belong to a recently emerged class of random-access algorithms, called "limited channel sensing" algorithms. Utilizing the regenerative character of the stochastic processes that are associated with the random access system, the authors derive lower bounds on the maximum stable throughput, and tight upper and lower bounds on the induced mean packet delay. The proposed algorithms are inherently stable, they combine good performance with modest channel sensing requirements, and they outperform their synchronous counterparts in some Ethernet and mobile radio environments.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: MAJ Brian W. Woodruff	22b. TELEPHONE NUMBER (Include Area Code): (202) 767- 5027; 22c. OFFICE SYMBOL: NM

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

HIGH PERFORMANCE ASYNCHRONOUS LIMITED SENSING ALGORITHMS FOR CSMA AND CSMA-CD CHANNELS

M. Georgiopoulos, L. Merakos
and P. Papantoni-Kazakos

EECS Department, U-157
University of Connecticut
Storrs, CT 06268

Abstract

We consider the random multiple access of a collision-type, packet-switched channel, for the Poisson user model in a local area network environment, where "carrier sensing" techniques are possible due to small propagation delays.

We propose and analyze asynchronous (unslotted) random access algorithms that belong to a recently emerged class of random-access algorithms, called "limited channel sensing" algorithms. Utilizing the regenerative character of the stochastic processes that are associated with the random access system, we derive lower bounds on the maximum stable throughput, and tight upper and lower bounds on the induced mean packet delay. The proposed algorithms are inherently stable, they combine good performance with modest channel sensing requirements, and they outperform their synchronous counterparts in some Ethernet and mobile radio environments.

Work supported by the Air Force Office of Scientific Research, under grant AFOSR-83-0229.

1. Introduction

Local area networks (LANs) are designed to support high bandwidth communications among a large number of users within a local geographical area. When a single channel (coaxial cable, fiber optic, or radio multi-access channel) is available, and the users are allowed to access the channel randomly (whenever they have something to transmit), a distributed control random access algorithm (RAA) may be used to coordinate the overall user activity. When the end-to-end propagation delay of the LAN is small, as compared to the transmission time of a packet, then the users can determine the channel activity in a short amount of time, through "channel sensing" operations.

The most well known RAAs for LANs belong to the class of "Aloha-type" algorithms, like the Non-persistent CSMA, CSMA/CD and their variations [1-4]. Although the above algorithms are easy to implement, they have inherent long-term stability problems. The recognition of this instability, by some researchers, led to the "Aloha-type with retrnasmission control" [5] and the "Tree-Search-type" [6-12] classes of algorithms. The algorithms in those two classes are stable, but they are inherently characterized by a major operational drawback. An integral part in their operation is the assumption that each user senses the channel continuously, even when he is not yet part of the system. This full channel sensing assumption equivalently implies that each user "knows" precisely, and at all times, the lag between the time instant, when the system started operating, and the time instant of his own arrival in the system. This extreme synchronization is clearly nonfeasible in most real systems, where activation of new users, user mobility, and actions of higher level protocols discarding packets already in the system [10,11], disrupt the feedback sensing continuity. In summary, the "Aloha-type" algorithms have poor performance due to instabilities, and the "Aloha-type with retransmission control" and the "Tree-Search-type" algorithms are nonimplementable, in most systems, due to extreme synchronization requirements.

In this paper, we consider the class of "Limited Channel Sensing" algorithms, [13-16, 21]. The algorithms in this class are stable, and they can attain high quality performance. In contrast to the "Tree-Search-type" algorithms, the "limited channel sensing" algorithms require that each user sense the channel, only from the time that he generates a packet, to the time that this packet is successfully transmitted (limited channel sensing). This level of channel sensing is clearly feasible in most LANs. We propose and analyze "Limited Channel Sensing" algorithms for the CSMA and CSMA-CD channels. In contrast to the algorithm in [18], the proposed algorithms operate asynchronously. That is, we assume that the users can not distinguish slot boundaries; thus, packet transmissions start at random times, rather than at the beginning of some slot. Asynchronous algorithms are of greater practical importance than synchronous algorithms, especially for local area networks, since there is no need for the users to maintain a global time base. In addition, such algorithms may out perform their synchronous counterparts, in some important local area environments (as we will see later).

Utilizing the regenerative character of the stochastic processes that are associated with the random access channel, we derive lower bounds on the maximum stable throughput, and tight upper and lower bounds on the induced mean packet delay. These results indicate that the proposed algorithms are inherently stable, and that they combine good performance with limited channel sensing and modest synchronization requirements.

2. User and Channel Model

We assume that an infinite population of identical, independent, bursty, packet-transmitting users share a common communication channel. We model the cumulative packet arrival process as homogeneous Poisson, with intensity λ packets per unit of time. We conveniently take the packet length to correspond to our unit of time. We also assume that the propagation delay between any two users in our system is at most α, where $\alpha \ll 1$. We note that the above Poisson user model best represents

environments, such as the packet radio and the Ethernet, where bursty independent users enter and depart the system randomly.

We consider limited channel sensing and ternary feedback. That is, each user senses the channel continuously, from the time instant when he generates a packet, to the time instant when this packet is successfully transmitted, and he can distinguish without error among the following channel states: a) idle (no transmission) b) success (transmission of a single packet) c) collision (simultaneous transmission of at least two packets). We assume that a collision results in complete loss of the information included in all the involved packets; thus, retransmission is then necessary. We assume that the users can not determine slot boundaries. We thus consider unslotted channel, and asynchronous transmissions. We initially introduce the following assumptions:

A1) The propagation delay between any two users in the network is exactly α.

A2) A collision is distinguished from a successful transmission, by all the users in the system (those who are then sensing the channel), if it lasts γ time units.

We point out that assumption A1 is initially adopted for analytical reasons. In practice, the propagation delay between any two users is a random variable, depending on the relative position of those two users in the network. We will discuss the relaxation of assumption A1 in section 6. Assumption A2 is realistic, where the length γ is a network characteristic determined by the encoding of the packets in baseband transmission, or by the modulation technique used in broadband transmission. Note that assumption A2 implies that, in addition to transmitting users, non-transmitting users also have the capability to detect collisions, provided that they are sensing the channel during the collision; (receive mode collision detection, IEEE Standards Committee, Project 802, [25]).

3. An Algorithm

We consider the user and channel model in section 2, and we adopt assumptions A1 and A2. We assume that all users know then the common propagation delay, α,

and the common parameter, γ, and we describe an asynchronous limited channel sensing algorithm implemented by each user in a distributed fashion. We name the algorithm, SALCS.

Each user has in memory the parameters, α and γ, and two common parameters, m and n, that are both positive integers, subject to optimization for throughput maximization. The time instant when the user generates a new packet, he immediately starts sensing the channel, and he simultaneously initializes the SALCS. He maintains continuous channel sensing until the successful transmission of his packet; upon the occurrence of this event, he stops sensing the channel. For the implementation of the SALCS, the user has a counter, whose value of time t is denoted r_t. The values of the counter dictate the operation of the algorithm, which is described as follows:

A. <u>Transmission rule</u>

The user transmits whenever his counter value equals "0".

B. *<u>Initial counter value</u>*

Let the user generate a new packet at time t_o. Then, he immediately starts sensing the channel, and he sets:

$$r_{t_o} = \begin{cases} 0 \; ; & \text{if he senses the channel idle at } t_o. \\ -1 \; ; & \text{if he senses the channel busy at } t_o. \end{cases}$$

C. <u>Special action after collision</u>

Let at time t the user start a transmission. Let t' be the first time instant after t that the user senses the channel state as collision, while still transmitting. Then,

a) If $t'-t \geq \gamma + \alpha$, the user extends his transmission to the time instant $t''=t+\gamma+2\alpha$. At t", he aborts his packet, and he considers t" as the time instant when the collision ends.

b) If $t'-t < \gamma + \alpha$, the user aborts his packet at t', and considers t' as the end of the collision.

D. <u>Recursions of counter value</u>

Let at time t the user sense the channel busy, and let his counter value be

then r_t. Let t' be the first time instant after t, that the user senses the channel idle. Then,

Rule 1: The counter value, r_t, remains unchanged, until the time instant t'.

Rule 2: At time t', the counter value is updated as follows.

a) If $r_t = -1$, then,

$$r_{t'} = M \; ; \; 0 \leq M \leq m-1, \text{ with probability } m^{-1}.$$

b) If at $(t')^-$ the user senses the channel state as success, and

$r_t \geq 1$, then,

$$r_{t'} = r_t + m-1$$

If $r_t = 0$, the user has been successfully transmitted at t'.

c) If at $(t')^-$ the user senses the channel state as collision, and

$r_t \geq 0$, then,

$$r_{t'} = \begin{cases} r_t+m+n-1 & ; \text{ if } r_t \geq 1 \\ m-1+N; 1 \leq N \leq n, \text{w.p. } n^{-1} & ; \text{ if } r_t = 0, \text{ and the user acted as in C. b) upon sensing the collision.} \\ -1 & ; \text{ if } r_t = 0, \text{ and the user acted as in C. a) upon sensing the collision.} \end{cases}$$

Rule 3: After time t', the counter value is updated as follows.

a) If $r_{t'} \geq 1$, the user decrements his counter value by one, every α time units, after t', to a value not less than 0, until the time instant t", that he first senses the channel busy again. Then, he maintains the counter value $r_{t''}$ unchanged, until he first senses the channel idle again. He then repeats rule 2.

b) If $r_t = 0$ and $r_{t'} = -1$, then,

$$r_\tau = -1 \; ; \text{ for } t' \leq \tau < t' + \alpha$$

$$r_{t'+\alpha} = m-1+N, \; 1 \leq N \leq n, \text{ with probability } n^{-1}.$$

The user decrements his counter value by one, every α time units, after $t'+\alpha$, to a value not less than 0, until the time instant t'', that he first senses the channel busy again. Then, he maintains the value $r_{t''}$ unchanged, until he first senses the channel idle again. He then repeats rule 2.

We note that in the SALCS, packets do not use the distinction between success and collision, until they first sense an idle channel state. This property actually represents an advantage, since this distinction may be impossible when the first sensed idle channel state appears shortly after arrival. From the operation of the algorithm, we conclude that the interval, $[t't'+\alpha)$, in part b) of rule 3 is necessarily idle. Also, due to assumption A2 in section 2, in conjuction with the SALCS operation, we conclude that a successful transmission lasts one time unit, while a collision lasts, $\beta \overset{\Delta}{=} \gamma + 2\alpha$, time units. The relationship between β, γ, and α is exhibited in figure 1.

The general operation of the SALCS is perhaps better illustrated, by introducing the concept of a "stack". A stack is an abstract storage device, consisting of an infinite number of cells, labelled -1, 0,1,2,3,... (figure 2). The number of packets that a cell can accommodate is unrestricted. At each time t during the operation of the algorithm, users with counter value r can be thought of as having stored their packets in cell "r" of the stack. A packet is transmitted whenever it enters cell "0" of the stack. Packets are eventually successfully transmitted after moving through the cells of the stack in accordance with the algorithmic rules described above.

4. SALCS Renewal Properties. Stability

Given the user and channel model in section 2, and adopting assumptions A1 and A2, the SALCS operates in sessions. The first session begins at time zero, when the system starts operating. The ith session begins at time R_i, and ends at time R_{i+1}, when the (i+1)th session begins; thus, $R_1 = 0$. To define the renewal time instants R_j, $j \geq 1$, we use the concept of a marker, that operates on the stack in figure 2. The marker is perceived as an outside observer, who knows the rules of

the SALCS, and who senses the channel continuously, with delay α. At time $R_1 = 0$, the marker is placed at cell #1. The marker stays at cell #1 until time t', when the channel changes state from busy to idle for the first time after R_1. From then on the marker moves through the cells of the stack according to the rules of the algorithm. At some time instant $t \geq t'$ (before the session ends), the marker is at cell # k of the stack, if from the algorithmic operations, in conjuction with the marker's channel sensing, it is deduced that cells #j, $j \geq k$ are necessarily empty at time t. R_2 occurs at some time instant t", if at time $t''+\alpha$ the marker is placed at cell #0. At R_2 the marker is automatically reset at cell #1, and the above process is repeated to determine R_3. This process continues indefinitely. Let us denote by L_i, the length of the ith session. From the above description of the sessions, in conjuction with the memoryless property of the Poisson arrival process, we then easily conclude that $\{L_i\}$ is a set of i.i.d. random variables. The sequence $\{R_n\}_{n \geq 1}$ forms then a renewal process, where,

$$R_1 = 0 \ , \ R_{n+1} = R_n + L_n \ , \ n=1,2,... \tag{1}$$

Let f(t) be a time function, such that f(t) = 1, during successful transmission periods, and f(t) = 0, otherwise. Let S_n denote the number of successful transmissions in the nth session, and let us define,

$$R(t) \triangleq \int_0^t f(t)dt \ , \ \sigma(t) \triangleq \frac{R(t)}{t} \tag{2}$$

$$S \triangleq E\{S_n\} \ , \quad L \triangleq E\{L_n\} \tag{3}$$

Given n, the random variable S_n depends on the random variable L_n, but the pairs (L_n, S_n), $n \geq 1$ are i.i.d. Also, due to (2) we have $S_n = R(R_n) - R(R_{n-1})$, and we can express the following theorem (see [17, thr. 3.6.1]).

<u>Theorem 1</u> If $S < \infty$ and $L < \infty$, then there exists a real number σ, such that $\lim_{t\to\infty} \sigma(t) = \lim_{t\to\infty} \frac{E(R(t))}{t} = \frac{S}{L} = \sigma$ with probability 1; the quantity σ is the channel's <u>output rate</u>.

Directly from the definition of a session, we conclude that all the arrivals generated while the session is in progress are successfully transmitted by its end. Thus, S_n also represents the number of arrivals during the nth session. Let X_1 denote the first, after the beginning of the nth session, time that an arrival occurs. Let X_i, $i \geq 2$, represent the interarrival time between the (i-1)th and the ith arrival, during the nth session. Let X denote the first time that an arrival occurs, after the end of the nth session. Then (see figure 3),

$$L_n + X = \sum_{i=1}^{S_n+1} X_i \tag{4}$$

The random variables, $\{X_i\}_{i\geq 1}$, are i.i.d., with common mean λ^{-1}. Also, if $L < \infty$, then $S < \infty$, where L and S are given by (3). Thus, assuming $L < \infty$, taking expected values in (4), and applying Wald's equation [17], we obtain, $L+\lambda^{-1}=\lambda(S+1)$, or,

$$S = \lambda L \; ; \text{ if } L < \infty \tag{5}$$

In view of (5), theorem 1 yields,

<u>Corollary 1</u>: If $L < \infty$, then $\sigma = \lambda$; that is the ALCS is then <u>stable</u>.

Thus, the algorithm maintains the rate, and it is thus stable, if the mean session length is finite. In the sequel, we investigate the conditions under which $L < \infty$, and we establish bounds on L.

a. <u>Mean Session Length. Throughput</u>

For evaluating the expected value of a session, we need the concept of a <u>subsession</u>. Unlike sessions, subsessions may be overlapping. A subsession of multiplicity $k \geq 1$ starts at time t if i) cell #0 of the stack is empty at t^-,

ii) cell #0 of the stack is nonempty at time t, and iii) cell #0 of the stack contains k packets at time $(t+\alpha)^-$. The end of a subsession of multiplicity k is determined via a marker. The marker (as in the case of a session) is perceived as an outside observer, who knows the rules of the SACLS, and who senses the channel continuously with delay α. At the beginning of a subsession of multiplicity $k \geq 1$, the marker is placed at cell #1, and stays there until the channel changes state from busy to idle for the first time after the beginning of the subsession. From then on the marker moves through the cells of the stack according to the rules of the algorithm. The subsession ends at some time $t'' > t$ if the marker is placed at cell #0 at time $t''+\alpha$.

Let L_k $(k \geq 1)$ denote the expected length of a subsession of multiplicity k, and let L be the expected length of a session in (3).

Consider now figure 4. Suppose that a session begins at time 0 and that the first arrival occurs at time t. It is then easy to see that,

$$L = \lambda^{-1} + \sum_{k=1}^{\infty} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k-1}}{(k-1)!} L_k \tag{6}$$

From expression (6) it is clear that to evaluate the expected length, L, of a session, we need to evaluate the expected lengths of a subsession of multiplicity k, $\forall\, k \geq 1$. The dynamics of the SALCS yield the following proposition, whose proof is in the appendix.

<u>Proposition 1</u> The expected lengths, $\{L_k\}_{k \geq 1}$, of subsessions satisfy an infinite dimensionality linear system of the form,

$$x_k = \sum_{\mu=1}^{\infty} A_{k,\mu}\, x_\mu + f_k \; ; \; k \geq 1 \tag{7}$$

; where the coefficients $\{A_{k,\mu}\}$and f_k are determined by the system characteristics, and are given in the appendix.

The following theorem specifies a sufficient condition under which system (7) has a solution, $\{y_k\}_{k\geq 1}$, with $0 \leq y_k < \infty$, for all $1 \leq k < \infty$, which coincides with the sequence, $\{L_k\}_{k\geq 1}$, of the expected subsession lengths induced by the algorithm.

Theorem 2

(i) Given α, β, $m \geq 1$, $n \geq 2$, there exist bounded functions of λ, b, b', c, c' and a positive number $\lambda_o(\alpha,\beta,m,n) < 1$ such that the system (7) has a solution, $\{y_k\}_{k\geq 1}$, such that,

$$0 < b'k-c' \leq y_k \leq bk-c \ , \ k \geq 1$$

if $\lambda < \lambda_o(\alpha,\beta,m,n)$.

(ii) Given α,β, given m,n, then, for every $\lambda\varepsilon(0,\lambda_o(\alpha,\beta,m,n))$, $L_k = y_k$, for all $k \geq 1$.

The coefficients b, b' and c, c' and the number $\lambda_o(\alpha,\beta,m,n)$ are derived in the proof of the theorem, which can be found in the appendix. Given the network parameters, α and β, let us now define,

$$\underline{\lambda}(\alpha,\beta) \stackrel{\Delta}{=} \sup_{m,n} \lambda_o(\alpha,\beta,m,n) = \lambda_o(\alpha,\beta,m^*,n^*) \qquad (8)$$

Directly from theorem 2 in conjuction with expressions (6) and (8), the following corollary evolves.

Corollary 2 Given α, β, the SACLS with operation parameters m^*, n^* is stable, for $\lambda\varepsilon(0,\underline{\lambda}(\alpha,\beta))$

We used numerical search techniques to determine the optimum values m^* and n^*, which maximize $\lambda_o(\alpha,\beta,m,n)$. In table 1 we give the values of m^*, n^* and $\underline{\lambda}(\alpha,\beta)$, for representative values of α and for $\beta = 2\alpha$ and $\beta = 1$. Given α and β, the number $\underline{\lambda}(\alpha,\beta)$ is a tight lower bound on the throughput of the SALCS, subject to assumptions A1 and A2. Throughput is here defined as the maximum Poisson rate maintained by the algorithm.

5. Delay Analysis

Let the arriving packets be labelled n=1,2,..., according to the order of their arrival instant. Let $\mathcal{D}_n$ denote the delay experienced by the nth packet (time interval between the time of its arrival and the time of its successful reception at its destination). Also, let M_i denote the total number of packets that were successfully transmitted during the first i sessions. The sequence $\{M_i\}_{i\geq 0}$ is a renewal sequence, since $M_0 = 0$, and $M_{i+1} = M_i + S_{i+1}$, $i \geq 0$; where, S_i, $i \geq 1$, represents the number of packets that were successfully transmitted during the ith session. From the renewal properties of the algorihtm, it can be seen that, for every $i \geq 0$, the process $\{\mathcal{D}_{M_i+n}\}_{n\geq 1}$, is a probabilistic replica of the process $\{\mathcal{D}_n\}_{n\geq 1}$. Thus, the process $\{\mathcal{D}_n\}_{n\geq 1}$ is regenerative with respect to the renewal sequence $\{M_i\}_{i\geq 0}$, with common regenerative cycle, S, the number of packets successfully transmitted over a session. The next theorem is a combination of theorem 2 and corollary 2 of [20].

<u>Theorem 3</u> If (A.1) S is not periodic, with $S = E(S) < \infty$, and if (A.2) $T \triangleq E[\sum_{i=1}^{S} \mathcal{D}_i] < \infty$, then there exists a parameter D such that $D = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathcal{D}_i = \lim_{n\to\infty} \frac{1}{n} E(\sum_{i=1}^{n} \mathcal{D}_i)$, with probability 1. Furthermore, $\mathcal{D}_n$ converges in distribution to a random variable $\mathcal{D}_\infty$, and

$$D = E(\mathcal{D}_\infty) = T/S < \infty \tag{9}$$

Consider again figure 4. Assume that a session starts at time 0, and that the first arrival occurs at time t. Let W_1 denote the expected cumulative delay experienced by all packets associated with a subsession of multiplicity one, <u>$1+\alpha$ units of time after</u> the beginning of the subsession. Let also $W_k (k>1)$ denote the expected cumulative delay experienced by all packets associated with a subsession of multiplicity k, <u>$\beta+\alpha$ units of time after</u> the beginning of the subsession. We

note that the packets associated with a session or a subsession are the correspondingly successfully transmitted packets. Let us now define,

$$T_1 = 1 + \alpha + \frac{\lambda}{2} + W_1 \tag{10}$$

$$T_k = \beta + \alpha + (k-1)(\beta + \frac{\alpha}{2}) + \frac{\lambda\beta^2}{2} + W_k \; ; \; k > 1 \tag{11}$$

Clearly,

$$T = \sum_{k=1}^{\infty} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k-1}}{(k-1)!} \; T_k \tag{12}$$

Due to the dynamics of the SALCS, and as in proposition 1, it can be shown that the set, $\{W_k\}_{k\geq 1}$, satisfies a linear system as that in (7). Then, a theorem parallel to theorem 2 can be expressed. Its proof is parallel to that of theorem 2, and it is omitted.

Theorem 4 Given α,β, given m^* and n^* as in (8), let the number $\lambda_o(\alpha,\beta,m^*,n^*) = \underline{\lambda}(\alpha,\lambda)$ be as in (8). Then, for every λ in $(0,\underline{\lambda}(\alpha,\beta))$ there exist positive bounded functions of λ, V_1, V_2, V_3, V_1', V_2', and V_3' (whose expressions can be found in [19]), such that,

$$V_1'k^2 + V_2'k + V_3' \leq W_k \leq V_1k^2 + V_2k + V_3 \; ; \; k \geq 1 \tag{13}$$

We note that unlike the bounds on the L_k's in theorem 2, the bounds on the T_k's are quadratic. Those bounds in conjuction with (9), (10), (11), (12), (5), (6), and the bounds in theorem 2, provide the means for the computation of upper and lower bounds on the expected per packet delay, D. Tighter bounds can be computed, however, via the method of truncated infinite linear systems, as in [21]. The specifics of the method for the present problem is omitted here, due to lack of space, and it can be found in [19]. We only include the computed, for various α and β choices, bounds. Those bounds can be found in table 2, where,

$$D_\ell \leq D \leq D_u \tag{14}$$

We note that the bounds D_ℓ and D_u remain extremely tight (actually they coincide to the fifth decimal point), even for λ values close to $\underline{\lambda}(\alpha,\beta)$. The expected per packet delay consists of two parts. The expected per packet access delay, which corresponds to the mean time required for a packet to access the nonnegatively indexed cells of the stack (cells #0,1,2 ...), and the expected per packet contention delay, which corresponds to the mean time elapsed from the moment a packet enters the nonnegatively indexed cells of the stack until it is successfully transmitted. The mean packet delay characteristics are perhaps better illustrated in figures 5 and 6, where we plot D_ℓ, D_u versus λ, along with the mean packet delays induced by the synchronous limited channel sensing algorithm developed in [18], for $\alpha = .1$, and $\beta = 1$ (i.e. $\gamma = .8$ for $\alpha = .1$) or $\beta = 2\alpha$ (i.e. $\gamma = 0$). We point out that $\gamma = 0$ corresponds to $\beta = 2\alpha$ for the SALCS, and it corresponds to $\beta = \alpha$ for the algorithm in [18], while $\gamma = .8$ corresponds to $\beta = 1$ for the SALCS and to $\beta = .9$ for the algorithm in [18]. We note that for the same values of α and γ and for small λ values ($\lambda < .25$), the expected per packet delay induced by the SALCS is smaller than the expected per packet delay induced by the synchronous algorithm in [18]. This is due to the fact that for small λ values, the expected per packet access delay induced by the former algorithm is smaller than the same delay induced by the latter algorithm, while the expected per packet contention delay is then almost the same for both algorithms. As λ increases, and the expected per packet contention delay becomes dominant (as compared to the expected per packet access delay), the synchronous algorithm out performs the SACLS. In environments where assumption A1 is true, but synchronization is difficult to achieve, this difference in performance may be more than off set by the ease in implementation of the asynchronous SALCS.

6. The Relaxation of Assumption A1

In sections 4 and 5, we analyzed the performance of the SALCS, subject to

assumption A1, in section 2. That is, we assumed that the propagation delay between any two users in the network is constant and identical, and it equals α time units. This assumption is in practice true, only for star networks. In mobile radio networks, and in LANs using a bus, as Ethernet does, the propagation delay between two arbitrary users is a random variable. The value of this variable depends on the relative distance between those two users in the network, and it takes values in $(0, \alpha]$, where α is here the network end-to-end propagation delay (i.e., the maximum propagation delay). The maximum propagation delay, α, is a network characteristic, and it is reasonable to assume that it is known by all users. Considering the SALCS whose operational parameter is this maximum propagation delay, α, the issue is: what are its performance characteristics, in the mobile radio and the Ethernet environments.

In the mobile radio environment, let some user, A, generate a new packet, and start implementing the SALCS until his packet is successfully transmitted. The relative to A positions of other users that interfere with A's transmissions, during the above period, change then randomly. This random change is best modelled by random propagation delays. Specifically, the actual propagation delay between user A and some other transmitting user is changing then every T time units, and each time it is a random variable, uniformly distributed in $[0, \alpha]$, where α is the network end-to-end propagation delay. The period T is determined by the mobility characteristics of the users; it is relatively small for highly mobile users, and it is relatively large for slowly moving users.

In the Ethernet environment, each user selects upon arrival his position on the line, randomly and uniformly. He remains in the same position until he departs the system. If α is the line end-to-end propagation delay, it is equivalent to say that the user selects upon arrival his propagation delay from a reference point (one end of the line), randomly and uniformly in $[0, \alpha]$, and that he maintains this property until he departs the system. Thus, the relative propagation delay between two users in the system equals then the difference between their propagation delays from the

reference point, it is strictly determined by the position of the users on the line, and it remains unchanged until the users depart the system.

We simulated the SALCS in the presence of the mobile radio and the Ethernet environments, for the Poisson user model, where each new packet is a different user. We selected representative values of the parameter γ, and of the maximum end-to-end network propagation delay; we denoted the latter, α. In both the mobile radio and the Ethernet simulations, we selected the maximum propagation delay, α, and $\alpha/2$, as the SALCS operational parameter. We assumed that the position of each user remains practically unchanged, from the time he generates a packet to the time this packet is successfully transmitted. This latter assumption reflects mobile radio LANs accurately, even for user velocities of the order of 3,000 miles/hour, due to the low transmission delays induced by the SALCS. Thus, for the same end-to-end propagation delay, α, our simulations are identical for both the Ethernet and the mobile radio LANs.

Our simulation results were practically identical, when the SALCS operational parameter is either the end-to-end propagation delay, α, or the mean propagation delay, $\alpha/2$. This represents a positive property of the SALCS; its performance is robust regarding the choice of the above operational parameter. It is thus sufficient that the users know only an approximate value of the network end-to-end propagation delay, for the SALCS implementation. In figure 7, we plot the simulated expected per packet delays, D, for varying Poisson rates, λ, together with the theoretically computed expected per packet delays induced by the SALCS, when assumption A1 holds (table 2). We observe that the SALCS performance is better in the Ethernet and mobile radio environments, than in a star network environment (where assumption A1 holds). This performance difference increases, as the network parameter, γ, decreases, reaching its maximum for $\gamma = 0$ (when a collision of practically zero length is distinguished, from a successful transmission). In addition, in the Ethernet and mobile radio environments, the SALCS out performs its synchronous counterpart ([18]), as concluded by comparing figures 5 and 6 with figure 7.

7. A Refined Algorithm

As we pointed out in section 3, in the SALCS, packets do not use the distinction between success and collision, until they sense the first idle channel state. The question that arises then is: If such a distinction is possible and used, will it result in significant performance improvement? To answer this question, we developed and analyzed a refined algorithm, named GALCS, adopting again assumptions A1 and A2, in section 2, The GALCS has the same channel sensing characteristics as the SALCS, and it is again implemented independently by each user via a counter. In addition to the parameters, α and γ, each user maintains in memory seven common parameters, m_1, m_2, n', p_o, q_o, r_o, and r_1, however. The parameters, m_1, m_2, and n' are positive integers, and the parameters, p_o, q_o, r_o, and r_1 are probabilities; they are all subject to optimization for throughput maximization. The operation of the GALCS is described by parts, A, B, and C, in the description of the SALCS (section 3), and by the recursions of the counter value, displayed below.

Recursions of counter value

Let at time t the user sense the channel busy, and let his counter value be then r_t. Let t' be the first time instant after t, that the user senses the channel idle. Then,

Rule 1: The counter value, r_t, remains unchanged until the time instant t'.

Rule 2: At time t' the counter value is updated as follows.
If at time $(t')^-$ the user senses the channel state as success and

a) $r_t = -1$, then,

$r_{t'} = 0$; with probability p_o

$r_{t'} = M$; $1 \leq M \leq m_1$, with probability $(1-p_o)\, m_1^{-1}$

b) $r_t \geq 1$, then,

$r_{t'} = r_t + m_1$

c) $r_t = 0$, then, the user has been successfully transmitted at t'.

If at $(t')^-$ the user senses the channel state as collision and

d) $r_t = -1$, then,

$r_{t'} = 0$; with probability q_o.

$r_{t'} = M$; $1 \leq M \leq m_2$, with probability $(1-q_o)\ m_2^{-1}$.

e) $r_t = 0$, and the user acted as in C. b) upon sensing the collision, then,

$r_{t'} = 0$; with probability r_o.

$r_{t'} = M$; $1 \leq M \leq m_2$, with probability $r_1 \cdot m_2^{-1}$.

$r_{t'} = m_2+N$; $1 \leq N \leq n'$, with probability $(1-r_o-r_1)n'^{-1}$

f) $r_{t'} = 0$, and the user acted as in C. a) upon sensing the collision, then,

$r_{t'} = -1$

g) $r_t \geq 1$, then

$r_{t'} = r_t + m_2 + n'$

Rule 3: After time t', the counter value is updated as follows.

a) If $r_{t'} \geq 1$, the user decrements his counter value by one, every α time units, <u>after t'</u>, to a value not less than 0, until the time instant t", that he first senses the channel busy again. Then, he maintains the counter value $r_{t''}$ unchanged, until he first senses the channel idle again. He then repeats rule 2.

b) If $r_t = 0$ and $r_{t'} = 1$, then,

$r_\tau = -1$; for $t' \leq \tau \leq t' + \alpha$.

$r_{t'+\alpha} = 0$; with probability r_o

$r_{t'+\alpha} = M$; $1 \leq M \leq m_2$, with probability $r_1\ m_2^{-1}$.

$r_{t'+\alpha} = m_2 + N$; $1 \leq N \leq n'$, with probability $(1-r_o-r_1)n'^{-1}$.

If $r_{t'+\alpha} \geq 1$, the user decrements his counter value by one, every α time units, <u>after t'+α</u>, to a value not less than 0, until the time instant t", that he first senses the channel busy again. Then he maintains the value $r_{t''}$ unchanged, until he first senses the channel idle again. He then repeats rule 2.

We observe that the SALCS evolves from the GALCS, via the following substitutions in the latter: $m_1=m_2=m-1$, $p_o=q_o=m^{-1}$, $r_o=r_1=0$, and $n'=n$. We note that the SALCS

represents a significant operational simplification over the GALCS. We analyzed the GALCS, subject to assumption A1. We found that the improvement in performance over the SALCS is then insignificant (throughput increase in the third decimal, in most cases), at the expense of increased operational complexity. The above statement is also true, in the presence of the mobile radio and the Ethernet-type environments. Thus, the SALCS is practically sufficient.

8. Conclusions. Comparisons

The existing classes of random access protocols for local area networks are reflected by references [3], [5], [12], [18], [22] and [23]. In [3], an "Aloha-type" protocol is studied. For the Poisson user model that we consider in this paper, the throughput (the maintainable Poisson intesity) of the above protocol is zero. The protocol proposed in [5] belongs to the "Aloha-type with retransmission control" class and it is stable, but it requires full channel sensing as well as knowledge of the system backlog, and it is synchronous (slotted channel). The protocol in [12] is basically Gallager's algorithm [9], with different cost assignments for empty, success, and collision channel states. This protocol also requires full channel sensing, and it is synchronous. The protocol in [22] is basically synchronous and it requires knowledge of the system backlog (thus complete knowledge of the total channel history).

Considering the fact that in most local area networks the users can not know the total channel history from the beginning of time (before they enter the system), the only sensible approach is the consideration of the "limited channel sensing" class of protocols. Those protocols only require that each user sense the channel continuously, from the time he generates a new packet, to the time this packet is successfully transmitted. The protocol in [18] belongs to this class; it requires, however, slotted channel and synchronous transmission. The protocol in [23] (ALCS) and the SALCS in this paper also belong to the limited channel sensing class, and they address the case where users can not distinguish slot boundaries, due to lack

of global time base. The ALCS and the SALCS are thus asynchronous, they are implemented by each new packet independently, and they induce minimal operational complexity. The difference between the ALCS and the SALCS protocols lies in the performance superiority of the latter. In star networks, where the propagation delay between any two users remains constant and unchanged, the SALCS performs worse than its synchronous counterpart. This difference in performance decreases monotonically to nil, as the propagation delay decreases to the order of thousandths of a packet length, and it is then compensated by the elimination of the synchronicity requirement in the SALCS. In mobile radio and Ethernet-type environments, the SALCS performs better than its synchronous counterpart, since it then models the real environment closer (see also [24]). The on y implementable in the above environments existing stable protocols are those in [18], in [23], and the SALCS. Among them, the SALCS is significantly superior.

References

[1] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I - Carrier sense multiple access models and their throughput-delay characteristics," IEEE Trans. Comm., vol. COM-23, pp. 1400-1416, Dec. 1975.

[2] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part IV - Stability considerations and dynamic control in carrier sense multiple access," IEEE Trans. Comm., vol. COM-25, pp. 1103-1119, Oct. 1977.

[3] F. A. Tobagi and V. B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," Comput. Nets, vol. 4, pp. 243-249, Oct. - Nov. 1980.

[4] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," Comm. Ass. Comput. Mach., vol. 19, pp. 395-403, July 1976.

[5] J. S. Meditch, and C. T. A. Lea, "Stability and optimization of the CSMA and CSMA/CD channels," IEEE Trans. Comm., vol. COM-31, No. 6, pp. 763-773, June 1983.

[6] J. F. Hayes, "An adaptive techniques for local distribution," IEEE Trans. Comm. vol. COM-26, pp. 1178-1186, Aug. 1978.

[7] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," IEEE Trans. Inform. Theory, vol. IT-25, pp. 505-515, Sept. 1979.

[8] B. S. Tsybakov and V. A. Mikhailov, "Slotted multiaccess packet broadcasting feedback channel," Problemy Peredachi Informatsii, vol. 14, no. 4, Oct/Dec. 1978.

[9] R. G. Gallager, "Conflict resolution in random access broadcast networks," presented at AFOSR Workshop on Commun. Theory, Provincetown, MA, 1978.

[10] J. L. Massey, "Collision resolution algorithms and random access communications," Multi-User Communications (CISM Courses and Lectures Series), G. Long, Ed., New York: Springer-Verlag, 1981

[11] M. L. Molle, "Unifications and extensions of the multiple access communications problem, "CSD Report No. 810730 (UCLA-ENG-8118), Computer Science Department, University of Califronia, Los Angeles (July 1981), Ph.D. Dissertation.

[12] D. Towsley and G. Venkatesh, "Window random access protocols for local computer networks," IEEE Trans. Comp., vol. C-31, pp. 715-722, Aug. 1982.

[13] B. S. Tsybakov and N. D. Vvedenskaya, "Stack algorithm for random multiple-access," Problemy Peredachi Informatsii, vol. 15, no. 3, pp. 80-94, Jul./sept. 1980.

[14] L. Merakos, "Limited Sensing random multiple access using binary feedback," in Proc. Conf. Inform. Sci. Syst., Baltimore, MD, March 1983, pp. 557-563.

[15] P. Mathys, and P. Flajolet, "Q-ary collision resolution algorithms in random access systems with free or blocked channel access," in Abst. Papers, 1983 IEEE Inform. Theory Symp., St. Jovite, Quebec, Canada, Sept. 1983.

[16] N. D. Vvedenskaya, and B. S. Tsybakov, "Random multiple access of packets to a channel with errors," Problemy Peredachi Informatsii, vol. 19, no. 2, pp. 52-68, Apr. - June 1983.

[17] S. M. Ross, Stochastic Processes, New York, Wiley 1983.

[18] L. Merakos, "Stack Algorithm for local area networks using carrier sense multiple access channels," submitted for publication; also Tech. Rep. TR-84-6, Dept. of Elec. Engrg. Comput. Sci. Univ. of Connecticut, Storrs, April 1984. Description of the algorithm included in (2d algorithm there) "Continuous entry random access algorithms for local area networks," Proceedings, GLOBECOM'84, pp. 19.4.1 - 19.4.6.

[19] M. Georgiopoulos, L. Merakos, and P. Papantoni-Kazakos, "An asynchronous stack algorithm for local area networks using carrier sense multiple access channels," Tech. Rep. TR-84-9, Dept. of Elec. Engrg. Comput. Sci., University of Connecticut, Storrs, August 1984.

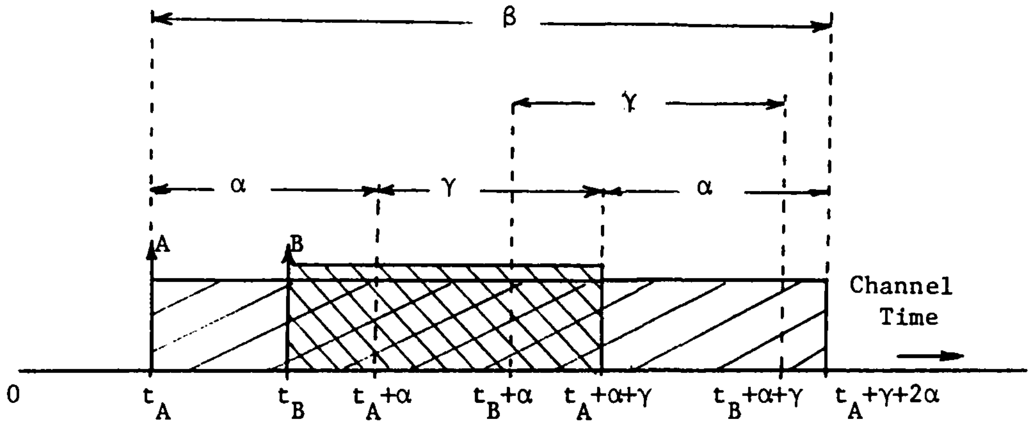
[20] S. Stidham, Jr., "Regenerative processes in the theory of queues, with applications to the alternating-priority queue," Adv. Appl. Probl., vol. 4, pp. 542-577, 1972.

[21] L. Georgiadis and P. Papantoni-Kazakos, "Limited feedback sensing algorithms for the broadcast channel," IEEE Trans. Inform. Th., Special Issue on Random Access Communications, to appear in March, 1985.

[22] I. Rubin, "Synchronous and channel-sense asynchronous diagram group-random-access schemes for multiple-access communications," IEEE Trans. Commun., vol. COM-31, pp. 1063-1077, Sept. 1983.

[23] M. Georgiopoulos, L. Merakos, and P. Papantoni-Kazakos, "An Asynchronous stack algorithm for CSMA and CSMA-CD channels," Proceedings, IEEE INFOCOM'85 conference.

[24] K. Sohraby, M. L. Molle, and A. N. Venetsanopoulos, "Why analytical models of Ethernet-Like local networks are so pessimistic," Proceedings, IEEE GLOBECOM'84, pp. 19.4.1-19.4.6.



A	: First arriving packet that participates in collision.
B	: A later than A arriving packet that collides with A.
t_A	: Arrival instant of packet A.
t_B	: Arrival instant of packet B.
$t_A+\alpha+\gamma$	: Time that B aborts transmission.
$t_B+\alpha+\gamma$	: Time that A senses collision.
$t_A+\gamma+2\alpha$	: Time that A aborts transmission.

Figure 1

Collision of two packets. SALCS Algorithm.

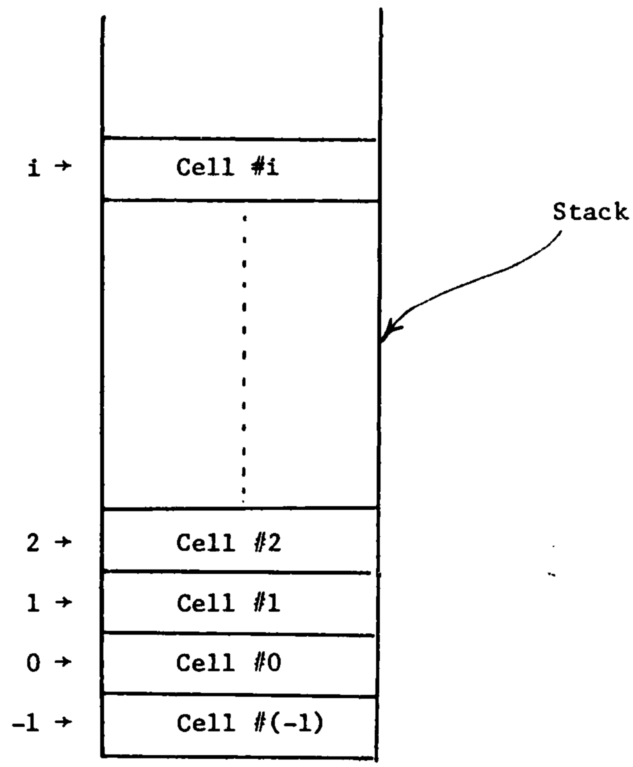


Figure 2

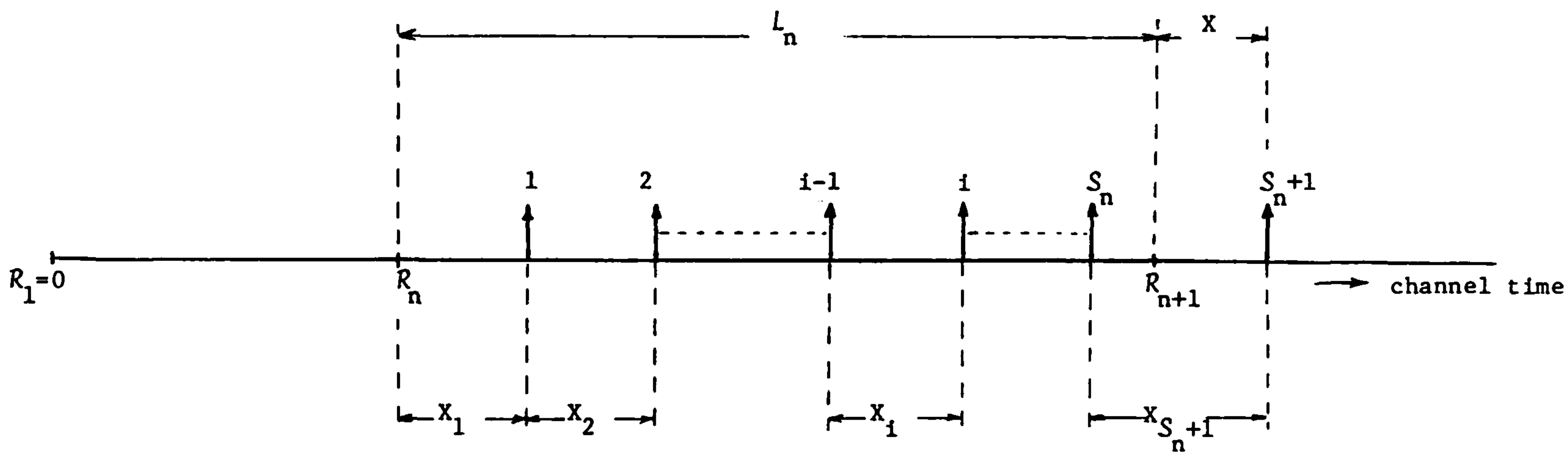
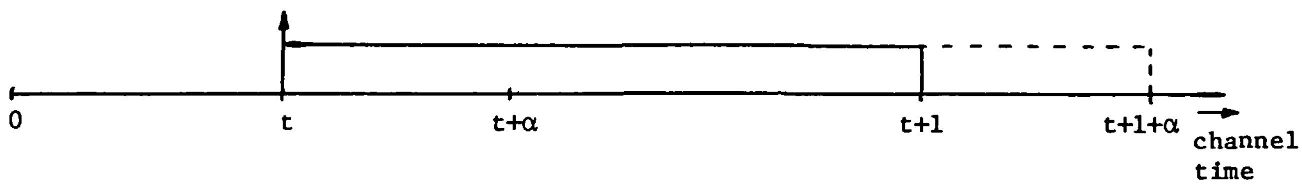


Figure 3

Case a



Case b

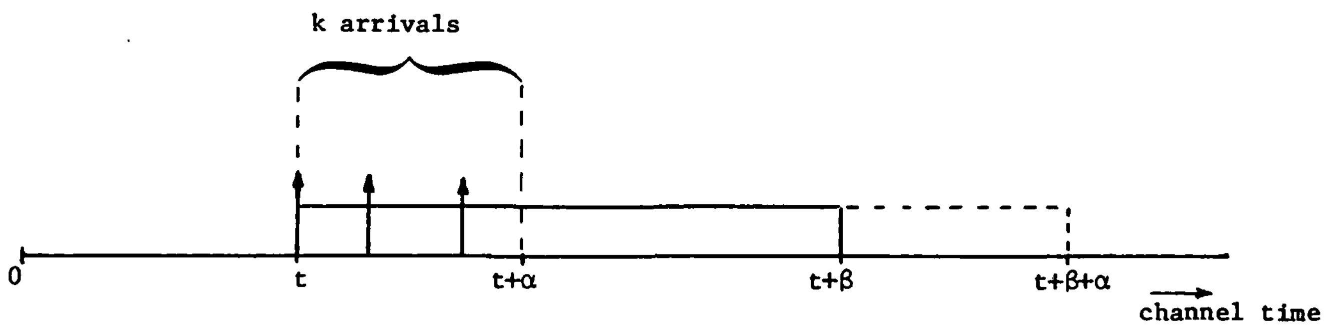


Figure 4

	$\beta = 2\alpha$		
α	$\underline{\lambda}(\alpha,\beta)$	m^*	n^*
.4	.346	1	4
.2	.530	1	4
.1	.699	1	4
.05	.824	1	3
.01	.958	2	3
.001	.995	2	4

	$\beta = 1.0$		
α	$\underline{\lambda}(\alpha,\beta)$	m^*	n^*
.4	.337	1	4
.2	.472	1	5
.1	.595	2	6
.05	.700	3	7
.01	.860	7	12
.001	.955	22	34

<u>Table 1</u>

SALCS Throughputs. Assumption A1 holding.

	$D_e \sim D_u$											
	$\beta = 1$						$\beta = 2\alpha$					
λ	$\alpha = .4$	$\alpha = .2$	$\alpha = .1$	$\alpha = .05$	$\alpha = .01$	$\alpha = .001$	$\alpha = .4$	$\alpha = .2$	$\alpha = .1$	$\alpha = .05$	$\alpha = .01$	$\alpha = .001$
.1	1.94258	1.42640	1.23463	1.14470	1.07572	1.05881	1.90306	1.37438	1.19381	1.12011	1.06808	1.05678
.2	3.39726	1.87379	1.46219	1.29077	1.16552	1.13298	3.19122	1.68505	1.33798	1.21905	1.14225	1.12666
.3	10.9843	2.90692	1.87629	1.52694	1.29251	1.23110	9.02114	2.29458	1.56641	1.36183	1.23975	1.21669
.332	27.0615	-	-	-	-	-	-	-	-	-	-	-
.341	-	-	-	-	-	-	25.1142	-	-	-	-	-
.4	-	6.49672	2.75350	1.94294	1.48124	1.36620	-	3.79913	1.95626	1.57731	1.37297	1.33703
.467	-	25.0949	-	-	-	-	-	-	-	-	-	-
.5	-	-	5.39317	2.80155	1.78345	1.56280	-	11.4037	2.72698	1.92888	1.56508	1.50598
.525	-	-	-	-	-	-	-	22.2193	-	-	-	-
.590	-	-	32.2172	-	-	-	-	-	-	-	-	-
.6	-	-	-	5.33141	2.32981	1.87294	-	-	4.84236	2.58697	1.86500	1.76037
.694	-	-	-	-	-	-	-	-	22.9450	-	-	-
.695	-	-	-	41.5347	-	-	-	-	-	-	-	-
.7	-	-	-	-	3.57289	2.43022	-	-	-	4.21875	2.39673	2.18679
.8	-	-	-	-	8.86014	3.71159	-	-	-	14.6263	3.59335	3.04863
.819	-	-	-	-	-	-	-	-	-	29.3971	-	-
.855	-	-	-	-	71.7569	-	-	-	-	-	-	-
.9	-	-	-	-	-	9.62608	-	-	-	-	8.75176	5.70730
.95	-	-	-	-	-	92.0000	-	-	-	-	-	-
.953	-	-	-	-	-	-	-	-	-	-	67.3284	-
.99	-	-	-	-	-	-	-	-	-	-	-	84.5761

Table 2

SALCS Delays. Assumption A1 holding.

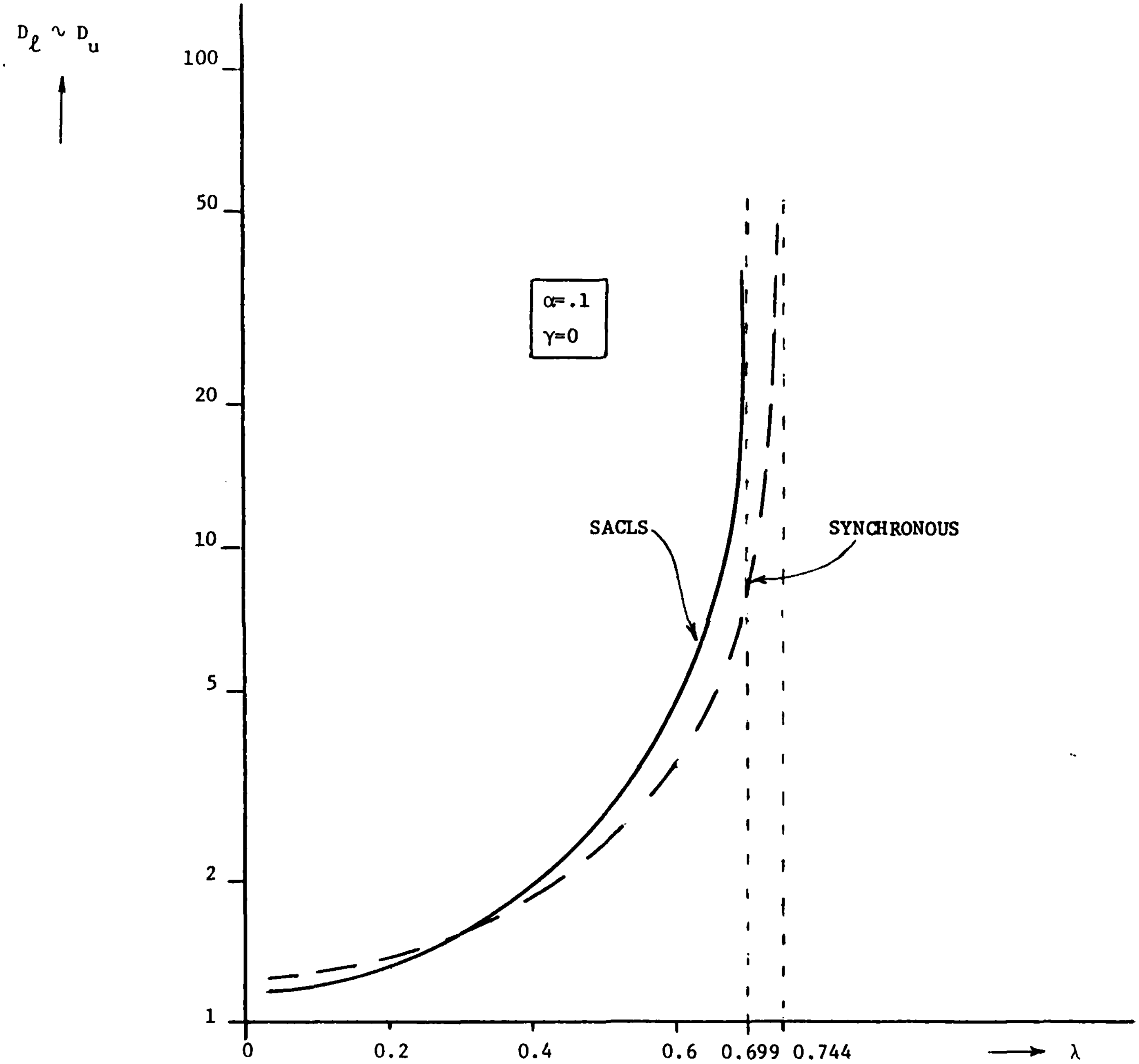
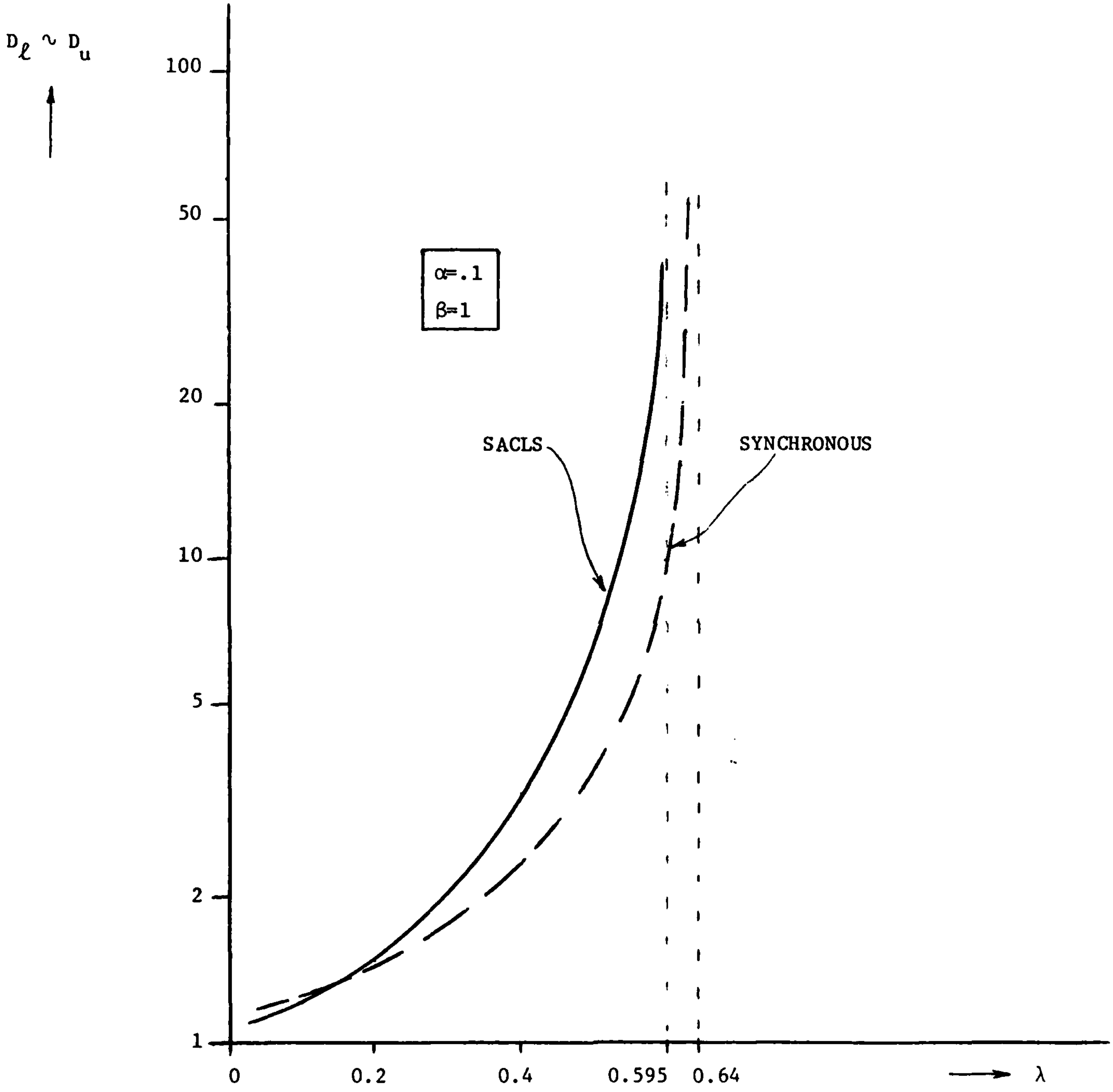


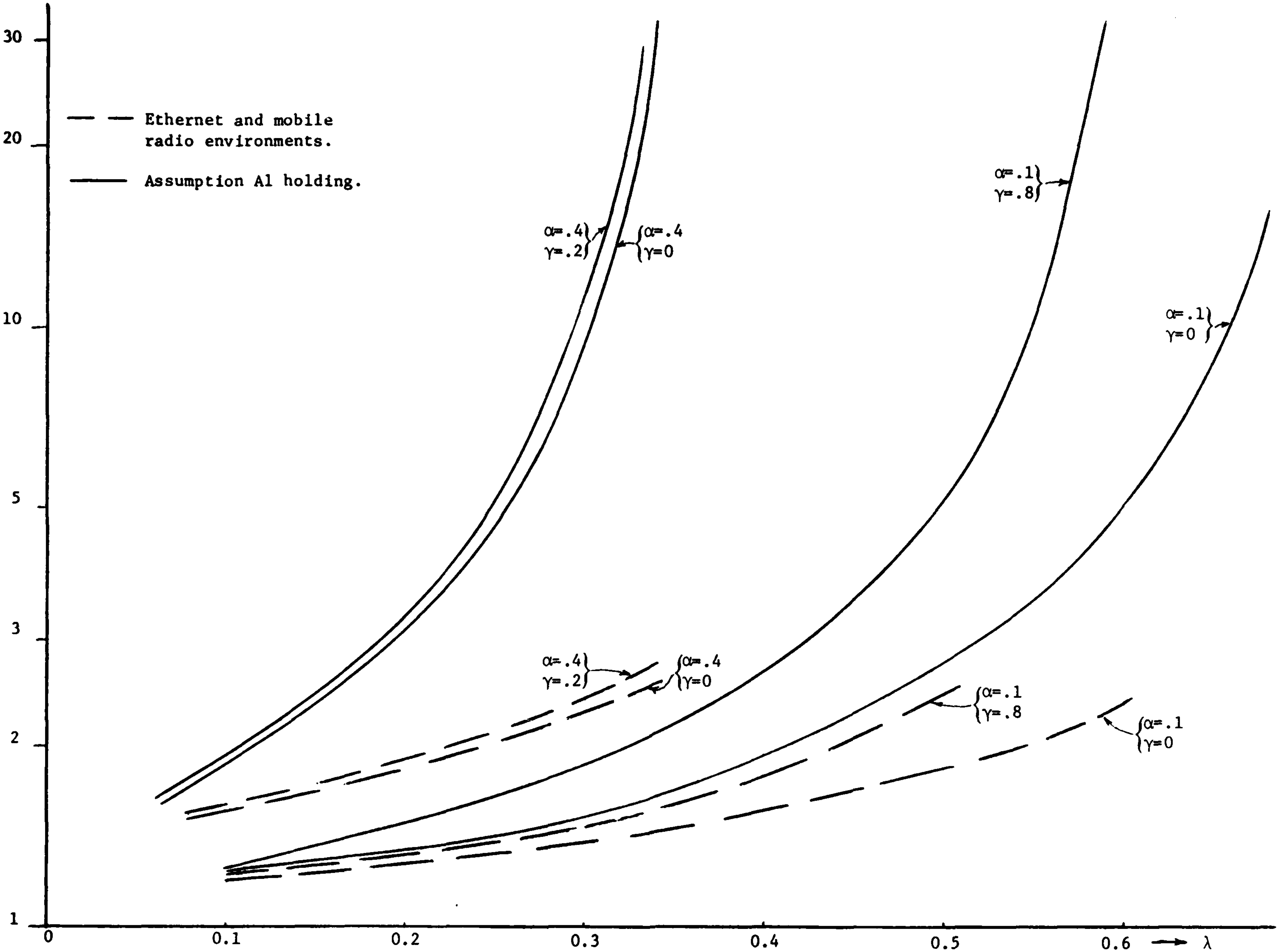
Figure 5

SALCS Delays. Assumption A1 holding.



<u>Figure 6</u>

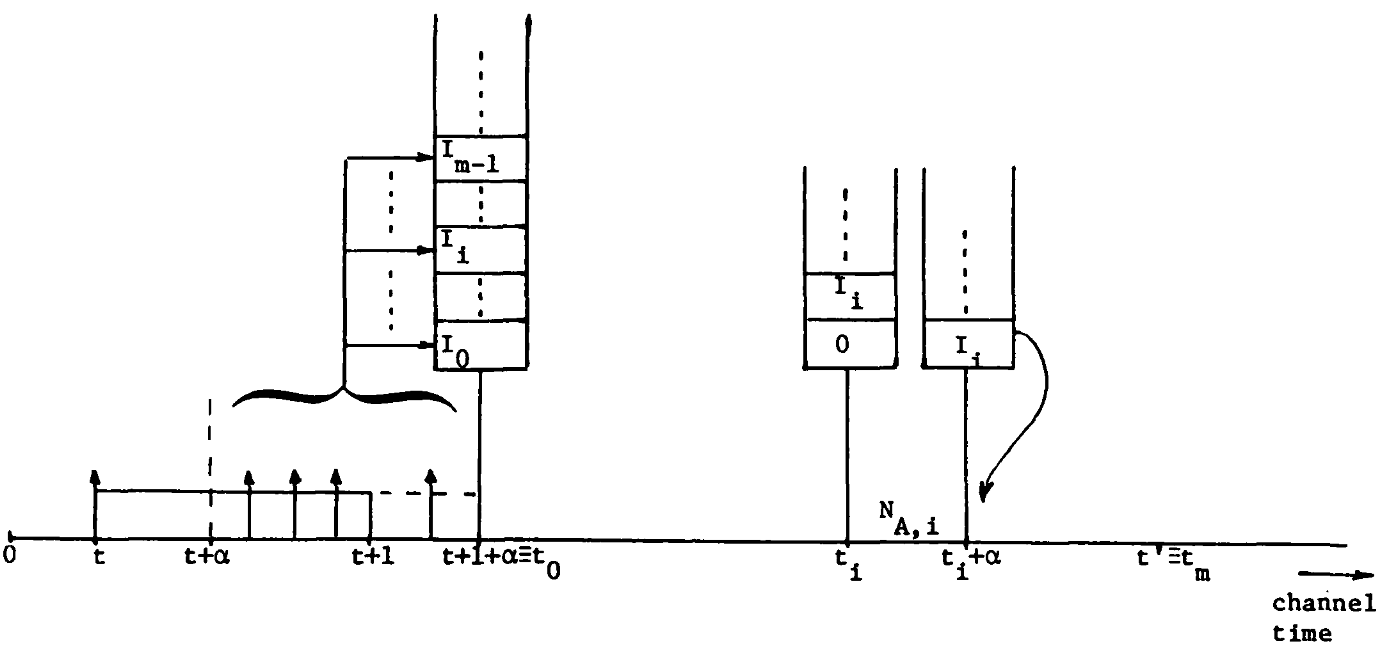
SACLS Delays. Assumption A1 holding.



30
20
10
5
3
2
1
Ethernet and mobile radio environments.
Assumption A1 holding.
α=.4 γ=.2
α=.4 γ=0
α=.1 γ=.8
α=.1 γ=0
α=.4 γ=.2
α=.4 γ=0
α=.1 γ=.8
α=.1 γ=0
0
0.1
0.2
0.3
0.4
0.5
0.6
λ

Figure 7

Case a



Case b

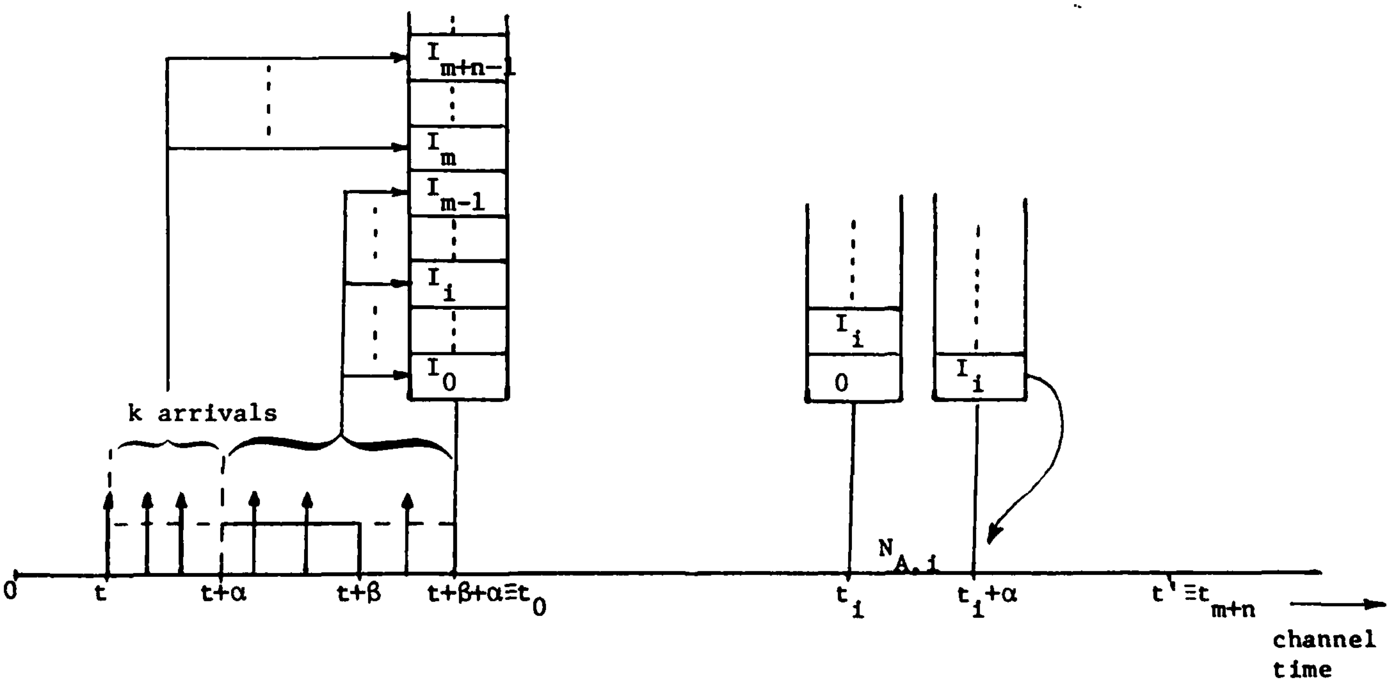


Figure 8

APPENDIX

Proof of Proposition 1

We will prove equation (7), for $k > 1$. The proof is similar, for $k = 1$. Assume that a session begins at time 0, and ends at time t'. (Figure 8). Let us also assume that the first subsession in the session is of multiplicity $k > 1$, and begins at time t (case b Figure 8). Let us denote by I_i; $0 \le i \le m+n-1$, the number of packets contained in cell #i of the stack at time $t+\beta+\alpha$. We denote by t_o the time $t+\beta+\alpha$, and by t_{m+n} the time t' (for uniformity of notation, which will become clear in the sequel). We denote by t_i ; $1 \le i \le m+n-1$ the first time, during the session, when the I_i packets occupy cell #1 of the stack, while cell #0 is empty. We can now define the random variable $t_{i,i-1}$; $1 \le i \le m+n$ as the difference between the r.v.'s t_i and t_{i-1}. That is, $t_{i,i-1} = t_i - t_{i-1}$; $1 \le i \le m+n$. Furthermore, $L'(I_o)$ denotes the expected value of $t_{1,0}$, $L''(I_{i-1})$ denotes the expected value of $t_{i,i-1}$ for $2 \le i \le m$, and $L'''(I_{i-1})$ denotes the expected value of $t_{i,i-1}$ for $m + 1 \le i \le m+n$. Then (see also figure 8) we have,

$$L_k = \beta+\alpha + L'(I_o) + U(m-1) \cdot \sum_{i=1}^{m-1} L''(I_i) + \sum_{i=m}^{m+n-1} L'''(I_i) \tag{A.1}$$

$$\text{where} \qquad U(m) = \begin{cases} 1 \text{ for } m > 0 \\ 0 \text{ otherwise} \end{cases}$$

We denote by $N_{A,i}$ $0 \le i \le m+n-1$ the number of new arrivals in the interval $(t_i, t_i+\alpha)$. We have also defined as L_k the expected length of a subsession of multiplicity k $(k \ge 1)$.

We now have,

$$L'(I_o) = L'(I_o|I_o=0)e^{-\lambda\beta/m} + L'(I_o|I_o > 0)(1-e^{\frac{-\lambda\beta}{m}}) \tag{A.2}$$

But,

$$L'(I_o|I_o=0) = 0 \tag{A.3}$$

while,

$$L'(I_o|I_o>0) = \left(1-e^{-\frac{\lambda\beta}{m}}\right)^{-1} \cdot \sum_{k_o=1}^{\infty} \sum_{k'_o=0}^{\infty} e^{-\frac{\lambda\beta}{m}} \frac{\left(\frac{\lambda\beta}{m}\right)^{k_o}}{k_o!} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k'_o}}{k'_o!} L_{k_o+k'_o} \tag{A.4}$$

Consequently (due to (A.2), (A.3), (A.4))

$$L'(I_o) = \sum_{k_o=1}^{\infty} \sum_{k'_o=0}^{\infty} e^{-\frac{\lambda\beta}{m}} \frac{\left(\frac{\lambda\beta}{m}\right)^{k_o}}{k_o!} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k'_o}}{k'_o!} L_{k_o+k'_o} \tag{A.5}$$

Furthermore,

$$L''(I_i) = L''(I_i|I_i=0)\, e^{-\frac{\lambda\beta}{m}} + L''(I_i|I_i>0)\left(1-e^{-\frac{\lambda\beta}{m}}\right);\ 1 \le i \le m-1 \tag{A.6}$$

Also,

$$L''(I_i|I_i=0) = L''(I_i|I_i=0,\ N_{A,i}=0)\, e^{-\lambda\alpha} + L''(I_i|I_i=0,\ N_{A,i}>0)\ (1-e^{-\lambda\alpha}) \tag{A.7}$$

But,

$$L''(I_i|I_i=0,\ N_{A,i}=0) = \alpha \tag{A.8}$$

while,

$$L''(I_i|I_i=0,\ N_{A,i}>0) = \frac{1-e^{-\lambda\alpha}-\lambda\alpha e^{-\lambda\alpha}}{\lambda(1-e^{-\lambda\alpha})} + \sum_{k'_i=0}^{\infty} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k'_i}}{k'_i!} L_{1+k'_i} \tag{A.9}$$

So,

$$L''(I_i|I_i=0) = \frac{1-e^{-\lambda\alpha}}{\lambda} + (1-e^{-\lambda\alpha}) \sum_{k'_i=0}^{\infty} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k'_i}}{k'_i!} L_{1+k'_i} \tag{A.10}$$

In a similar way we can show that,

$$L''(I_i|I_i>0) = \frac{1-e^{-\lambda\alpha}}{\lambda} + \qquad ;\ 1 \le i \le m-1$$

$$+ \left(1-e^{\frac{-\lambda\beta}{m}}\right)^{-1} \cdot \left\{ e^{-\lambda\alpha} \sum_{k_i=1}^{\infty} \sum_{k_i'=0}^{\infty} e^{-\frac{\lambda\beta}{m}} \frac{\left(\frac{\lambda\beta}{m}\right)^{k_i}}{k_i!} \; e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{k_i+k_i'} + \right.$$

$$\left. + (1-e^{-\lambda\alpha}) \sum_{k_i=1}^{\infty} \sum_{k_i'=0}^{\infty} e^{\frac{-\lambda\beta}{m}} \frac{\left(\frac{\lambda\beta}{m}\right)^{k_i}}{k_i!} \; e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{k_i+k_i'+1} \right\} \qquad \text{(A.11)}$$

As a result,

$$L''(I_i) = \frac{1-e^{-\lambda\alpha}}{\lambda} + e^{\frac{-\lambda\beta}{m}} (1-e^{-\lambda\alpha}) \sum_{k_i'=0}^{\infty} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{1+k_i'} + \qquad 1 \le i \le m-1$$

$$+ e^{-\lambda\alpha} \sum_{k_i=1}^{\infty} \sum_{k_i'=0}^{\infty} e^{\frac{-\lambda\beta}{m}} \frac{\left(\frac{\lambda\beta}{m}\right)^{k_i}}{k_i!} \; e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{k_i+k_i'} +$$

$$+ (1-e^{-\lambda\alpha}) \sum_{k_i=1}^{\infty} \sum_{k_i'=0}^{\infty} e^{\frac{-\lambda\beta}{m}} \frac{\left(\frac{\lambda\beta}{m}\right)^{k_i}}{k_i!} \; e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{k_i+k_i'+1} \qquad \text{(A.12)}$$

Following a similar procedure we can show that

$$L'''(I_i) = (1 - \tfrac{1}{n})^{k} \cdot (1-e^{-\lambda\alpha}) \sum_{k_i'=0}^{\infty} e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{1+k_i'} + \frac{1-e^{-\lambda\alpha}}{\lambda} \qquad m \le i \le m+n-1$$

$$+ e^{-\lambda\alpha} \sum_{k_i=1}^{k} \sum_{k_i'=0}^{\infty} \binom{k}{k_i} \left(\frac{1}{n}\right)^{k_i} (1 - \tfrac{1}{n})^{k-k_i} \; e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{k_i+k_i'} +$$

$$+ (1-e^{-\lambda\alpha}) \sum_{k_i=1}^{k} \sum_{k_i'=0}^{\infty} \binom{k}{k_i} \left(\frac{1}{n}\right)^{k_i} (1 - \tfrac{1}{n})^{k-k_i} \; e^{-\lambda\alpha} \frac{(\lambda\alpha)^{k_i'}}{k_i'!} \; L_{k_i+k_i'+1}$$

(A.13)

Let us now define $p(x,k) \triangleq e^{-x}\frac{x^k}{k!}$ and $b_{k'}(k,n) \triangleq \binom{k}{k'}\left(\frac{1}{n}\right)^{k'}(1-\frac{1}{n})^{k-k'}$

$x_1 \triangleq \frac{\lambda}{m}$, $x_2 = \frac{\lambda\beta}{m}$, $x_3 \triangleq \lambda\alpha$

Then (A.1), (A.5), (A.12) and (A.13) imply that

$$f_k = \beta + \alpha + (m+n-1)\,\frac{(1-e^{-\lambda\alpha})}{\lambda} \tag{A.14}$$

$$\begin{aligned}
A_{k,\mu} = {} & \sum_{\substack{\sigma=1 \\ \sigma+k=\mu}}^{\infty}\sum_{k'=0}^{\infty} p(x_2,\sigma)\cdot p(x_3,k') + \\
& + U(m-1)\cdot(m-1)\cdot\Bigg\{e^{-x_2}(1-e^{-x_3})\cdot p(x_3,\mu-1) + e^{-x_3}\cdot\sum_{\substack{\sigma=1 \\ \sigma+k'=\mu}}^{\infty}\sum_{k'=0}^{\infty} p(x_2,\sigma)\; p(x_3,k') + \\
& + (1-e^{-x_3})\sum_{\substack{\sigma=1 \\ \sigma+k'=\mu-1}}^{\infty}\sum_{k'=0}^{\infty} p(x_2,\sigma)p(x_3,k')\Bigg\} + \\
& + n\cdot\Bigg\{(1-\frac{1}{n})^{k}\,(1-e^{-x_3})\cdot p(x_3,\mu-1) + e^{-x_3}\sum_{\substack{\sigma=1 \\ \sigma+k'=\mu}}^{k}\sum_{k'=0}^{\infty} b_\sigma(k,n)\cdot p(x_3,k') + \\
& + (1-e^{-x_3})\sum_{\substack{\sigma=1 \\ \sigma+k'=\mu-1}}^{k}\sum_{k'=0}^{\infty} b_\sigma(k,n)p(x_3,k')\Bigg\}\;;\; k > 1
\end{aligned} \tag{A.15}$$

When k = 1 we take (working in a similar way as when k > 1) that

$$f_1 = 1 + \alpha + (m-1)\,\frac{(1-e^{-\lambda\alpha})}{\lambda} \tag{A.16}$$

$$\begin{aligned}
A_{1,\mu} = {} & \sum_{\substack{\sigma=1 \\ \sigma+k'=\mu}}^{\infty}\sum_{k'=0}^{\infty} p(x_1,\sigma)\; p(x_3,k') + \\
& + U(m-1)\;(m-1)\Bigg\{e^{-x_1}\,(1-e^{-x_3})\;p(x_3,\mu-1) + e^{-x_3}\sum_{\substack{\sigma=1 \\ \sigma+k'=\mu}}^{\infty}\sum_{k'=0}^{\infty} p(x_1,\sigma)\; p(x_3,k') +
\end{aligned}$$

$$+ (1-e^{-x_3}) \sum_{\substack{\sigma=1 \\ \sigma+k'=\mu-1}}^{\infty} \sum_{k'=0}^{\infty} p(x_1,\sigma)\, p(x_3,k') \Bigg\} \tag{A.17}$$

Proof of Theorem 2

Part (i). For convenience, we define for $X = \{x_k\}_{k\geq 1}$,

$$0_k(X) = \sum_{\mu=1}^{\infty} A_{k,\mu}\, x_\mu + f_k \;;\; k \geq 1 \tag{A.18}$$

The system in (7) can be then expressed as $x_k = 0_k(X)$. Given constants b, c we construct a sequence, $Z^{(0)}=\{z^{(0)}\}_{k\geq 1}$, such that $z_k^{(0)}=bk-c$; $k \geq 1$. After straightforward but tedious algebra we obtain,

$$0_k(Z^{(0)}) = x_k^{(0)} + g_k \;;\; k \geq 1 \tag{A.19}$$

; where,

$$g_k = f_k + f_{kb} \cdot b - f_{kc} \cdot c \;;\; k \geq 1 \tag{A.20}$$

and,

$$\left.\begin{aligned} f_{1b} &= -1 + \lambda + \lambda\alpha(1-e^{-\lambda/m}) + (m-1)(1-e^{-\lambda\alpha})(1+\lambda\alpha) + (m-1)e^{-\lambda\alpha}(1-e^{-\lambda/m})\lambda\alpha \\ f_{1c} &= -1 + m(1-e^{-\lambda/m}) + (m-1)e^{-\lambda/m}(1-e^{-\lambda\alpha}) \\ f_{kb} &= \lambda\beta + \lambda\alpha(1-e^{-\lambda\beta/m})+(m+n-1)(1-e^{-\lambda\alpha})(1+\lambda\alpha) + (m-1)e^{-\lambda\alpha}(1-e^{-\lambda\beta/m})\lambda\alpha + \\ &\quad + ne^{-\lambda\alpha}[1-(1-\tfrac{1}{n})^k]\lambda\alpha \\ f_{kc} &= m(1-e^{-\lambda\beta/m}) +(m-1)e^{-\lambda\beta/m}(1-e^{-\lambda\alpha}) + (n-1)[1-e^{-\lambda\alpha}(1-\tfrac{1}{n})^{k-1}] \end{aligned}\right\} \tag{A.21}$$

We now define the sequence, $Z^{(n)} = \{z_k^{(n)}\}_{k\geq 1}$; $n \geq 1$, such that, $z_k^{(n)} \triangleq 0_k(Z^{(n-1)})$. Since $0_k(\cdot)$ is nonnegative it follows that for every $k \geq 1$, the sequence $\{z_k^{(n)}\}_{n\geq 0}$ is nonnegative and nonincreasing, if b, c are such that,

$$bk-c \geq 0 \;,\; \forall\; k \geq 1 \quad \text{and} \quad g_k \leq 0 \;,\; \forall\; k \geq 1 \tag{A.22}$$

Thus, given the conditions in (A.22), the following limit exists.

$$\lim_{n\to\infty} z_k^{(n)} = z_k \leq bk-c \ ; \ k \geq 1 \qquad (A.23)$$

The sequence, $Z = \{z_k\}_{k\geq 1}$, solves system (7). Careful examination of conditions (A.22) reveals that if we choose b, c, such that, $g_1 = g_2 = 0$, then the rest of conditions (A.22) are satisfied provided that,

$$r(\lambda,\alpha,\beta;m,n) = \frac{f_{2c}\ f_{1b}}{f_{1c}} - f_{2b} > 0 \qquad (A.24)$$

As a result, $\lambda_o(\alpha,\beta;m,n)$ is the unique solution of the equation $r(\lambda,\alpha,\beta;m,n) = 0$. Furthermore,

$$b = \left[f_2 - \frac{f_{2c}\ f_1}{f_{1c}} \right] \cdot \left[\frac{f_{2c}\ f_{1b}}{f_{1c}} - f_{2b} \right]^{-1} \qquad (A.25)$$

and,

$$c = \frac{f_1}{f_{1c}} + \frac{f_{1b}}{f_{1c}}\ b \qquad (A.26)$$

The construction of the lower bounds in theorem 2 is similar to that of the upper bounds, and, therefore, it is omitted. We only give below, the expressions for b' and c'.

$$b' = \left[f_\infty - \frac{f_{\infty c}\ f_1}{f_c} \right] \cdot \left[\frac{f_{\infty c}\ f_{1b}}{f_{1c}} - f_{\infty b} \right]^{-1} \qquad (A.27)$$

where, $f_\infty = \lim_{k\to\infty} f_k$, $f_{\infty b} = \lim_{k\to\infty} f_{kb}$, $f_{\infty c} = \lim_{k\to\infty} f_{kc}$

and,

$$c' = \frac{f_1}{f_{1c}} + \frac{f_{1b}}{f_{1c}}\ b' \qquad (A.28)$$

<u>Part (ii)</u> In this part we show that the expected length, L_k, induced by the algorithm, coincides with the solution $Z = \{z_k\}_{k\geq 1}$ of part (i). First we prove the following lemma.

Lemma A The solution of system, $x_k = 0_k(X)$, is unique in the class of nonnegative sequences, $Y = \{y_k\}_{k\geq 1}$, which satisfy the condition,

$$\lim_{k\to\infty} \frac{y_k}{k^2} = 0 \tag{A.29}$$

Proof Assume that the system, $x_k = 0_k(X)$ has two nonnegative solutions satisfying condition (A.29), and let the sequence $D = \{d_k\}_{k\geq 1}$ denote their difference. Then, D satisfies (A.29) and solves the following system,

$$d_k = \sum_{\mu=1}^{\infty} A_{k,\mu}\, d_\mu \quad ; \; k \geq 1$$

For $u_1 > 0$ and $u_1 k^2 + u_2 k + u_3 > 0 \;\forall\; k \geq 1$, let us define a sequence $W = \{w_k\}_{k\geq 1}$, such that $w_k = d_k/(u_1 k^2 + u_2 k + u_3)$; $k \geq 1$.

Then the sequence W solves the following system,

$$w_k = \sum_{\mu=1}^{\infty} \bar{A}_{k,\mu}\, w_\mu \tag{A.30}$$

; where $\bar{A}_{k,\mu} = A_{k,\mu}/(u_1 k^2 + u_2 k + u_3)$; $k \geq 1$, $\mu \geq 1$. If we choose $u_2 = b$, $u_3 = -c$, then it can be shown that there exists $u_1 > 0$, such that,

$$\sum_{\mu=1}^{\infty} \bar{A}_{k,\mu} < 1 \text{ , for every } k \geq 1 \tag{A.31}$$

Furthermore we know that $w_k \to 0$ as $k \to \infty$. We will now show that the homogeneous system (A.30) cannot have a solution tending to zero and different from zero. Let us denote by $Q > 0$ the exact upper bound of $|w_k|$. Since $w_k \to 0$, we will have, beginning with some $k(k \geq k_1)$, $|w_k| < Q/2$, and therefore it must be possible to find a $k_o < k_1$ such that $|w_{k_o}| = Q$. From (A.30), for $k = k_o$, we have,

$$Q = |w_{k_o}| \leq \sum_{y=1}^{\infty} |\bar{A}_{k_o,y}|\ |w_y| \leq Q \sum_{y=1}^{\infty} |\bar{A}_{k_o,y}| \tag{A.32}$$

From (A.31) and (A.32), we have that $Q = 0$, and therefore $w_k = 0$, for every $k \geq 1$. Thus $d_k = 0$, for every $k \geq 1$, and the proof of the lemma is complete.

The solution Z of part (i) clearly satisfies condition (A.29). Thus, from lemma A, we have that Z is the unique nonnegative solution to system (7), satisfying condition (A.29). Next we use arguments similar to those used in Theorem 2 of [16] to show that $L_k = z_k$, for every $k \geq 1$. Let us consider the random variables $\ell_k(\tau) = \min(\ell_k, \tau)$, $k \geq 1$ where ℓ_k is the subsession length of multiplicity k, and τ is a real number, $\tau > 0$. Also let $L_k(\tau) = E\{\ell_k(\tau)\}$; $k \geq 1$. It can be easily seen that, for every $k \geq 1$:

$$L_k(\tau_1) \leq L_k(\tau_2) \qquad \text{if} \qquad 0 < \tau_1 \leq \tau_2 \tag{A.33}$$

$$\lim_{\tau \to \infty} L_k(\tau) = L_k \tag{A.34}$$

From (A.33) and from the dynamics of the algorithm, we clearly have that,

$$L_k(\tau) \leq f_k + \sum_{y=1}^{\infty} A_{k,y}\ L_y(\tau)\ , \text{ for every } k \geq 1 \text{ or}$$

$$L_k(\tau) = f_k - \overline{f_k} + \sum_{y=1}^{\infty} A_{k,y}\ L_y(\tau)\ ;\ \overline{f_k} \geq 0 \text{ for every } k \geq 1. \tag{A.35}$$

In addition, the sequence $\{L_k(\tau)\}_{k \geq 1}$ satisfies condition (A.29) for every τ, since $L_k(\tau) \leq \tau$. In view of (A.35), this fact implies that $L_k(\tau)$ is equal to the difference of the nonnegative solutions z_k and $\overline{z_k}$ of system (7) that respectively correspond to the forcing terms f_k and $\overline{f_k}$. That is, $L_k(\tau) = z_k - \overline{z_k}$, and therefore $L_k(\tau) \leq z_k$ for

every $\tau > 0$. Finally, taking limits as $\tau \to \infty$, we have, from (A.34), that $L_k \le z_k$, for every $k \ge 1$. Thus, the sequence $\{L_k\}_{k \ge 1}$ satisfies condition (A.29), and in view of lemma A we have $L_k = z_k$, for every $k \ge 1$.

END

FILMED

6-85

DTIC